Peer Review - Bullet

Kajsa Bjäräng, Viktoria Enderstein, Elin Eriksson, Lisa Fahlbeck, Alice Olsson
2018-10-23

The program as a whole and the thoughts behind it feels thought-through and well structured. The group behind the project seems to have a clear picture of the application they want to create, and they feel it is an application that is needed and wanted by many.

The first impression of the code is very overwhelming due to the applications many packages and packages within packages, where some only contain a single class or interface. This however may be due to plans of future expansion where more classes will be added with more functionality. Be that as it may, the number of packages currently appears unnecessary and makes it difficult to navigate the code. The classes themselves seem very structured and thought-through with very clear names.

In many places in the code there are calls to methods that don't exist, and similarly, there are many methods that have been written but are never used. Our guess it that the group has not yet finished their implementation, and that the application will work and look very differently after this week has passed. This, together with incomplete RAD and SDD make it slightly difficult for an outsider to presently completely understand the code.

The design model gives a clear overview of the application, even though the amount of packages and classes naturally take time to understand. It is however not updated in relation to the code, which makes it difficult to translate between the two. There are many classes, but specifically many packages missing in the design model so far. If the design model had been updated it would probably have been a very good tool in understanding the code in more detail.

As the application is very extensive there is understandably a large amount of dependencies, even some inbetween packages. These are however mainly unidirectional and necessary. Large parts of the model are also wholly independent and reusable. For example the two packages Services and Utils, which both contain functionality which easily can be implemented in other projects.

The documentation of the code is inconsistent. Some classes are void of documentation while some are explained well through javadoc. The package "bullet" is relatively well-documented, but many of the other packages are so far partially lacking in the documentation aspect.

The application follows multiple design patterns and design principles. For example, the class *Bullet* works as a facade for the application. The MVC pattern is followed by dividing the application into six different packages, “bullet”, “controller”, “view”, “util”, “services” and “content”. Bullet is the model of the application. The view package consists of multiple packages inside, and is responsible for taking input from the user and also for handling the graphical representation. The Factory pattern is also used to hide internal implementation. An example is BulletPointTreeFactory and SerializerFactory. The Observer pattern is also used multiple times in the view to give dynamic response to the user.

The tests implemented in the project are easy to find in a separate package called “test”. We assume this part of the project is not completed yet, as the coverage of the tests is so far not optimal. The “Bullet” package has a coverage of 28% of the class, and 12% of it’s methods. The other packages have similar test-coverage.

It is not possible to test the GUI completely, as the view does not seem to be finished and/or implemented and connected yet. From the tests and from running the code we can see that most methods are functional, but they are so far not implemented graphically in the view.

The level of encapsulation in the code is inconsistent and could probably be improved upon. Many public methods could probably instead be set to package-private, and some to private.

In conclusion, Bullet as an application seems to be well thought-through and the developers seem to be both ambitious and capable. It seems the implementation is not completely done yet, and the application will probably seem more complete next week. There are naturally shortcomings, but this is understandable, and overall the application has great potential.